AFRL-IF-RS-TR-2002-156
Final Technical Report
July 2002

HIGH PERFORMANCE VIRTUAL MACHINES

University of Illinois

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J089

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-156 has been reviewed and is approved for publication.

APPROVED: 

EDWARD DEPALMA
Project Engineer

FOR THE DIRECTOR: 

MICHAEL L. TALBERT, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2002	Final Aug 00 – Oct 01

4. TITLE AND SUBTITLE
HIGH PERFORMANCE VIRTUAL MACHINES

5. FUNDING NUMBERS
C - F30602-96-1-0286
PE - 62301E
PR - D985
TA - 01
WU - 01

6. AUTHOR(S)
Andrew A. Chien

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Illinois
Grants and Contracts Office
109 Coble Hall
801 S. Wright Street
Champaign Illinois 61820-6242

8. PERFORMING ORGANIZATION REPORT NUMBER
GCC0Q387

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFTB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-156

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Edward DePalma/IFTB/(315) 330-3069/ Edward.Depalma@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

The primary goal of the High Performance Virtual Machines (HPVM) project was to reduce the effort required to build a high performance cluster and distributed applications by leveraging the investments and understanding of scalable parallel systems. The approach to reducing the programming effort required to build high performance distributed applications was to insulate the program with a uniform, portable abstraction -- a High Performance Virtual Machine -- with predictable, high performance characteristics. Success was achieved by delivering a large fraction of the underlying hardware performance, visualizing resources to provide portability and reduce the application building effort and delivering predictable, high performance computing. HPVM has produced several major software releases involving high performance communication libraries, complete cluster software systems, and improved versions of those systems on a variety of hardware and software platforms. These systems have been downloaded and deployed at top research universities, major computer companies and national research laboratories. Clusters of commodity systems connected by high-speed networks are an important computing element and serve as an important model for future computing environments. Technologies which effectively exploit distributed computational resources can tap this "cluster pool" to deliver high performance computing, dramatically increasing the computational power available for both high performance computing and high performance distributed applications.

14. SUBJECT TERMS
Computational Clusters, Distributed Application Computing, High Performance Computing, Distributed Computing Environment

15. NUMBER OF PAGES
14

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table Of Contents

PROJECT GOALS	**1**
KEY RESEARCH RESULTS	**2**
STUDENTS AND STAFF	**4**
SOFTWARE	**5**
PAPERS	**6**
TECHNOLOGY TRANSFER	**9**

Project Goals

The rapidly increasing performance of low-cost computing systems has produced a rich environment for desktop, distributed, and wide-area computing. However this proliferating wealth of computational resources has not yet been effectively harnessed for high performance computing and high performance distributed applications such as scientific computing, signal processing, high speed interaction with ensembles of fixed scientific instruments, early detection sensor arrays, and immersive, interactive virtual reality environments (CAVE's) for C^3I. The performance potential of the proliferating computational infrastructure is staggering—billions of instructions per second and gigabytes of memory in a typical office. In many organizations, these vast computational resources are underutilized, idling much of the day and all of the night. **Technologies which effectively exploit distributed computational resources can tap this resource pool to deliver high performance computing, dramatically increasing the computational power available for both HPC and high performance distributed applications.**

Inexpensive, high speed networks are the critical enabling hardware technology for exploiting distributed computational resources, dramatically reducing the time for data shipping and coordination of parallel elements. Commodity commercial networks achieve 50-160MByte/s or more (e.g. Myrinet , Tandem's Servernet, Digital's Memory Channel, and emerging Infiniband implementations). At the time the project was begun, less aggressive networks were already widely deployed (OC-3c ATM at 18MByte/s and fast ethernet at 12MByte/s).

While high speed networks are still a rarity, the rapid progress of network technology ensures that high bandwidth will be ubiquitous in the future. High performance switched network environments not only provide a vehicle for exploiting distributed computational resources, they are also a good model for more tightly planned high performance network installations of today and tomorrow such as a embedded local-area networks on an AEGIS cruiser or within an aircraft. In addition, high performance switched network environments also present a potential model of the future computing systems, where bandwidth on demand services enable the configuration of high performance intranets (based on ATM or some alternative technology) to connect distributed computing resources. In short, **clusters of commodity systems connected by high speed networks are an important computing element in their own right, but also serve as an important model for future computing environments in terms of their scale of parallelism, high speed of network communication, and increasingly commodity software environments.**

The basic impetus behind high performance virtual machines is to exploit the software tools and developed understanding of parallel computation on scalable parallel systems to program clusters and distributed computing resources. In scalable parallel systems, application programmers rarely manage myriad distributed systems details—the network is an integral part of the machine, and resources are presented in a regularized form. The

performance model varies modestly from machine to machine, but the key elements of locality, communication, and sequential performance are constants. As a result, it is still dramatically easier to build applications, and portable, high performance applications are beginning to emerge. **The primary goal of high performance VM's was to reduce the effort to build high performance cluster and distributed applications by leveraging our investment and understanding of scalable parallel systems.** This effort will enable distributed resources to be easily tapped for high performance scientific computing and novel distributed applications.

Key Research Results

Our approach to reducing programming effort required to build high performance distributed applications was to insulate the program with a uniform, portable abstraction—a ***High Performance Virtual Machine*** -- with predictable, high performance characteristics. We succeeded in these goals by (1) delivering a large fraction of the underlying hardware performance, (2) virtualizing resources to provide portability and to reduce the effort in building application programs, and (3) delivering predictable, high performance. In a series of software releases, we have demonstrated viable solutions to these problems and further distributed working implementations which were used in production. In particular, we achieved the following:

- Research and development of low-level messaging abstractions which enable the delivery of the underlying hardware performance for a wide range application programming interfaces and across a wide range of the underlying hardware. These are the Fast Messages interfaces and associated guarantees.
- Delivery of a series of low-level messaging layer implementations, Fast Messages (FM), which provided the key guarantees which enabled the delivery of network hardware performance up to the applications. In particular, the FM layer delivered 70-90% of the underlying bandwidth with minimum transmission latencies within a small multiple of the hardware latency. For example, bandwidths of 100 – 200 MB/s second were achieved on underlying hardware with limits close by. Latencies of a few microseconds were achieved in networks where the underlying hop latency was approximately a microsecond.
- The abstractions implemented by Fast Messages (reliable, ordered message delivery) provided a portable communication abstraction across a number of distinct hardware and operating system platforms (multiple versions of both Windows and Linux), allowing application software to be moved across those platforms with little or no change. Based on the realized premise that FM was delivering all of the underlying performance, application retuning was generally not necessary. FM was implemented on the Cray T3D, two generations of Myrinet Networks, and the Tandem Servernet network.

- Research and development of high level messaging abstraction implementations which enable the delivery of the underlying low-level messaging performance
- Delivery of higher level messaging/communication interface implementations including, Messaging Passing Interface (MPI), Shmem Put/Get, Global Arrays, and BSP (bulk-synchronous programming) which each delivered a large fraction of the underlying hardware performance. Implementation of a range of higher level messaging/communication interfaces not only demonstrated the generality and flexibility of the interface, the effort to do so produced insights which led to real improvements in the FM interface and implementation.
- Delivery of turnkey cluster computing software packages which included communication/messaging, job scheduling, and performance monitoring. Assembling and integrating the key elements of software required for a complete commodity cluster system allowed easy construction and use of high performance virtual machines (HPVM's) which in turn increased the usability and utility of the abstractions for a range of software application tools.

Students and Staff

The following students were supported on this contract:

- Brian Fin, MS 1997 (employed at Hewlett Packard)
- Mario Lauria, MS 1997, PhD University of Naples, 1998 (faculty at Ohio State University)
- Steve Hoover, MS 1997 (employed at Digital / Compaq Computer)
- Matt Buchanan, MS 1998 (employed at Compaq Computer)
- Louis Giannini, MS 1999
- Geetanjali Sampemane, MS 2000 (continuing PhD Student)
- Scott Pakin, PhD 2001 (currently seeking employment)

The following staff was supported on this contract:

- Greg Koenig (now NCSA programmer staff)
- Qian Liu (now NCSA programmer staff)
- Philip Papadopoulous (now SDSC programmer staff)
- Greg Bruno (now SDSC programmer staff)
- Caroline Papadopoulos (now UCSD Computer Science support staff)
- Mason Katz (now SDSC programmer staff)

The students and staff have gone on to a wide range of industrial and academic opportunities, bringing with them substantial knowledge of clusters, and the key results about fast communication in cluster systems. They are now important catalysts for the transfer of high performance cluster technology both in industrial sites (Compaq, Digital, Hewlett-Packard, etc.) as well as in the major NSF national computing centers (NCSA and SDSC). For example, Philip Papadopoulos is now the leader for the clusters effort "ROCKS" at SDSC, and Scott Pakin is involved in the architecting of a communication layer for the "TeraGrid", a $56 million NSF-funded effort to build a cluster of clusters spread across the wide-area between Illinois and San Diego.

Software

The HPVM project has produced several major software releases involving high performance communication libraries, complete cluster software systems, and improved versions of those systems on a variety of hardware and software platforms. These systems have been downloaded and deployed at hundreds of sites around the world, including the top research universities, major computer companies, national research laboratories, etc. The major software releases, functionality, and release dates are summarized below.

- **Fast Messages 1.1 for Windows NT4.0 and Linux 2.x (released May 1997).** Included a basic high performance messaging layer, Fast Messages that ran on both Sockets and Myrinet, and, providing a capability to both prototype/develop software for the Fast Messages and run it with high performance. This system supported both Windows and Linux Systems. Peak 70MB/s of the Fast Messages layer on this platform was significantly greater than that achieved by other messaging layers.
- **HPVM 1.0 for Windows NT4.0 and Linux 2.x on x86, including FM 1.1 and MPI-FM 1.1 (released August 1997).** Included a high performance application messaging implementation (MPI) and a basic high performance messaging layer, Fast Messages that ran on Myrinet, and, providing a capability to both prototype/develop software for the Fast Messages and run it with high performance. This system supported both Windows and Linux Systems. Subsequently, a minor release included interface software for a commercial job queueing system, Platform's Load-sharing Facility (LSF) to be integrated into the system. Peak performance of 80MB/s for FM and 70+ MB/s for MPI were the fastest communication performance available on commodity hardware for 12 to 18 months from this release date.
- **HPVM 1.0 for Linux 2.x on x86, including FM 1.1 and MPI-FM 1.1 (maintenance release May 1998).** Support for changes in the underlying Linux 2.x platform.
- **Fast Messages 1.1 for Windows NT4.0 and Linux 2.x (source code May 1998).** Release of the source code, updated for the maintenance release in order to shared the support effort of the underlying system.
- **HPVM 1.2 for Windows NT on x86 (released 20 January 1999).** Included a high performance application messaging implementation (MPI), a basic high performance messaging layer, Fast Messages that ran on Myrinet, and an interface software for a commercial job queueing system, Platform's Load-sharing Facility (LSF) to be integrated into the system.

- **HPVM 1.9 for Windows NT on x86 (release 20 August 1999).** Significant improvements included more robust performance, peak performance of 100MB/s between boxes, and 200MB/s within SMP boxes. Included a high performance application messaging implementation (MPI), a basic high performance messaging layer, Fast Messages that ran on Myrinet, and an interface software for a commercial job queueing system, Platform's Load-sharing Facility (LSF) to be integrated into the system.

Papers

1996 and earlier

- *High Performance MPI Implementation on a Network of Workstations* Lauria's M.S. thesis, Oct 96 (Lauria)
- *High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet* In Supercomputing '95, San Diego, California, (Pakin, Lauria & Chien)

1997

- *Coordinated Thread Scheduling for Workstation Clusters Under Windows NT*. Proceedings of USENIX Windows NT Workshop, August 1997 (Buchanan & Chien).
- *Dynamic Coscheduling for Workstation Clusters*. Submitted for Publication (Sobalvarro, Pakin, Weihl & Chien), March 1997.
- *High Performance Virtual Machines (HPVM): Clusters with Supercomputing APIs and Performance*. Eighth SIAM Conference on Parallel Processing for Scientific Computing (PP97); March, 1997. (Chien, Pakin, Lauria, Buchanan, Hane, Giannini, and Prusakova)
- *Fast Messages (FM): Efficient, Portable Communication for Workstation Clusters and Massively-Parallel Processors*. IEEE Concurrency, vol. 5, no. 2, April-June 1997, pp. 60-73. (Pakin, Karamcheti & Chien)
- *MPI-FM: High Performance MPI on Workstation Clusters* Journal of Parallel and Distributed Computing, Vol. 40, No. 1, pp. 4-18, January 1997. (Lauria & Chien)

1998

- *A Software Architecture for Global Address Space Communication on Clusters: Put/Get on Fast Messages*. Proceedings of the 7th High Performance Distributed Computing (HPDC7) conference (Chicago, Illinois), July 28-31, 1998. (Giannini & Chien).
- *Efficient Layering for High Speed Communication: Fast Messages 2.x*. Proceedings of the 7th High Performance Distributed Computing (HPDC7) conference (Chicago, Illinois), July 28-31, 1998. (Lauria, Pakin & Chien).
- *Dynamic Coscheduling on Workstation Clusters*. Proceedings of the International Parallel Processing Symposium (IPPS '98), March 30-April 3, 1998. (Sobalvarro, Pakin, Chien & Weihl).

1999

- Performance Enhancements for HPVM in Multi-Network and Heterogeneous Hardware. Proceedings of PDC annual conference, December, 1999. (Bruno, Chien, Katz, and Papadopoulos).
- Feedback-based Synchronization for QoS Traffic in Cluster Computing. Proceedings of the sixth international conference on Parallel Interconnects(PI'99, formally known as MPPOI), October, 1999. (H. J. Song and A. Chien)
- FM-QoS: A Quality of Service Messaging Substrate for Asynchronous Local-Area Networks with Hardware-Level Network Feedback. (Word Doc) Kay Connelly's M.S. thesis, May 1999 (Connelly)
- Performance Evaluation of a Hardware Implementation of VIA. (Tech Report, Xin Liu)
- *A High Speed Disk-to-Disk sort on a Windows NT cluster running HPVM*. (Rivera & Chien).
- *Design and Evaluation of an HPVM-based Windows NT Supercomputer*. The International Journal of High-Performance Computing Applications, Vol. 13, No. 3, Fall 1999, pp. 201-219. (A. Chien, M. Lauria, R. Pennington, M. Showerman, G. Iannello, M. Buchanan, K. Connelly, L. Giannini, G. Koenig, S. Krishnamurthy, Q. Liu, S. Pakin & G. Sampemane).
- *Efficient Layering for High Speed Communication: the MPI over Fast Messages (FM) Experience*. Cluster Computing 2 (1999), pp. 107-116. (M. Lauria, S. Pakin, A. Chien).

2000

- Performance Monitoring on an HPVM Cluster Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, June 2000 (G. Sampemane, S. Pakin, Chien)
- Disk-to-disk Parallel Sorting on HPVM Clusters Running Windows NT Luis Rivera's M.S. Thesis, January 2000
- Design and Implementation of a Cluster Messaging Layer: Illinois Fast Messages on Servernet Louis Giannini's M.S. Thesis, January 2000

2001

- An HPVM Performance Monitor Geetanjali Sampemane's M.S. Thesis, January 2001

Technology Transfer

The HPVM project has successfully pursued a multi-channel approach to technology transfer.

Publish Papers: As documented elsewhere in this report, we have published numerous papers in leading conferences which document the advanced capabilities and key technologies developed by the HPVM project.

Distribute Software: As documented elsewhere in this report, we have distributed a series of working prototypes, all of which were widely used, in many cases in production or near production settings. These projects constitute an embodiment of the technical advances developed by the HPVM project.

Build large-scale systems: We have worked with computing centers in a number of sites to help them to build large-scale clusters based on HPVM. These include laboratories in the Department of Energy, abroad, and in the two NSF-funded supercomputing centers. In particular, we worked with the National Center for Supercomputing Applications to build an NT Supercluster, a 512 processor cluster with a capability of over 300 gigaflops, running HPVM 1.2.

Participate in Standards and Influence Industry Direction: The Fast Messages project is a cited contributor to the Virtual Interface Architecture, a cluster communication standard defined cooperatively by Intel, Compaq, and Microsoft, with the participation of over 100 industry vendors. Fast Messages was influential in a number of ways in the definition of the VIA standard, including:

- Descriptor formats which include immediate data (as well as pointers to buffers of data). The Fast Messages project demonstrated the importance of short messages and how they must be efficiently supported using immediate data and processor mediated I/O. This is a critical feature for NICs for high speed networks. This feature (immediate data) shows up in the control segment and provides the basis for efficient "linkage" for short transfers and short reads.

- Simple FIFO queue management was demonstrated as critical for short message performance by Fast Messages and a key feature of VIA for simplifying the underlying NIC hardware while maintaining high performance for short messages.

- Remote Direct Memory Access (RDMA). This feature was introduced in the ASCI Red machine but never enabled (bugs in the hardware). The viability and utility of this interface for clusters was demonstrated as part of the FM and HPVM projects and utility of this interface both in Shmem Put/Get interface and Global Arrays interfaces. In addition, Chien's work with Tandem contributed to these features in Tandem's Servernet which also drove the inclusion of these features in VIA.

- Reliability attributes. The Fast Messages' studies on the impact of reliability overhead on performance (ASPLOS IV, 1994) combined with Tandem's concerns led to the inclusion of reliability attributes (and the opportunity for network software adaptation at higher levels) in the VIA specification.

The Virtual Interface Architecture was defined and accepted by a large collection of vendors in 1998. Products based on this standard were introduced by Myricom, Compaq, Giganet, Finisar, Dolphin, and several others. However, in 1999 the VIA standard was combined with a cluster communication standard for input/output. This combination greatly slowed the ultimate definition of a cluster communication standard, and the resulting combined standard, called Infiniband, is only just now (Fall 2001) beginning to see products introduced. However, Infiniband effectively includes most of the key technical features of the Virtual Interface Architecture. Some of the software techniques used in Fast Messages were incorporated in a higher performance version of the Winsock (Microsoft's socket communication interface) Direct Interface, which shipped in the Windows 2000 DataCenter version, the 2nd Quarter of 2000.